DOCUMENT RESUME

ED 298 728 EC 210 764

AUTHOR Hedbring, Charles
TITLE Empirical Software Evaluation: A Practical Alternative.
PUB DATE Dec 87
NOTE 16p.
AVAILABLE FROM Hedbring and Associates, Suite 1712 Master Apartments, 310 Riverside Dr., New York, NY 10025.
PUB TYPE Journal Articles (080) -- Tests/Evaluation Instruments (160)
JOURNAL CIT Australian Journal of Human Communication Disorders; v15 n2 p75-88 Dec 1987

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *Computer Software; *Courseware; Educational Technology; Elementary Secondary Education; *Evaluation Criteria; Instructional Effectiveness; Media Selection; Microcomputers; Nonprint Media; *Severe Disabilities

ABSTRACT

The article presents a software evaluation checklist developed by a teaching-research laboratory for severely handicapped students in New York City. In an introductory section, the use of laptop microcomputers in helping handicapped learners acquire, maintain, and generalize functional skills is described as the fifth ingredient of an integrated curriculum approach featuring established conditions of learning, effective principles of teaching, proven rules of instruction, and community-referenced curricula. The 25-item checklist, with accompanying elaborative narrative, addresses the following topics: compatibility, media, backup disk, target population specification, returns policy, longterm cost, empirical software, load speed, introductory credits, attention screens, friendliness, failsafe operation, fault-tolerance, on-line help, flexible content, one-pass introduction, program filters, task versus reward frames, type of reward frames, levels of difficulty, auto-branching, on-line curriculum aids, automatic record-keeping, keyboard control, and alternative input options. Brief appendices present the checklist in abbreviated form and list selected software programs that rate highly on the checklist. (JW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED298728

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Charles Hedbring

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

EC 210 764

2

EMPIRICAL SOFTWARE EVALUATION: A PRACTICAL ALTERNATIVE

Charles Hedbring, Director
Program STEPPL, New York City

The purpose of this article is to present the STEPPL Software Evaluation Checklist that is used as a screening instrument in a teaching-research computer laboratory for severely handicapped learners located in New York City. The article first outlines the four components that comprise a contemporary instructional technology. Next, the importance of "laptop" microcomputers as the fifth component of the instructional revolution is discussed. The article then sets forth a twenty-five item checklist developed in the STEPPL Lab to select educational software for use with this new technology. Finally, a number of software programs that rate highly on the checklist are recommended for users of the Apple or IBM computer.

A mature instructional technology exists for teaching the learning handicapped. Trained personnel have at their disposal an integrated curriculum approach that includes four key features: (1) established conditions of learning (Bijou, 1972; Gagne, 1977; Kazdin, 1978); (2) effective principles of teaching (Bijou, 1979; Bloom, 1976; Gage, 1978; Lovaas, 1981); (3) proven rules of instruction (Bricker & Bricker, 1974; Engelmann & Carnine, 1982; Glaser, 1976; Guess, Sailor & Baer, 1976; Guess & Siegel-Causey, 1985; Haring & Schiefelbusch, 1976; Sailor et al., 1980; Sailor, Wilcox & Brown, 1980; Snell, 1983; Taba, 1962; Tyler, 1949); and (4) community-referenced curricula (e.g., Adams, 1975; Bender & Valletutti, 1976; Fredericks et al., 1976; Hedbring & Newsom, 1985; Kent, 1974; Kissinger, 1981; Popovich, 1977, 1981; Popovich & Laham, 1981; Sontag, 1977; Tawney, 1979; VORT, 1980).

The recent advent of microcomputer technology has added a fifth ingredient to the curriculum package. Specifically, the arrival of highpowered, durable, and lightweight "laptop" or "carry-along" computers assures that functional skills trained in the classroom can now be transferred to post-school work, play, and living environments (Brown et al., 1976; Stokes & Baer, 1977; White, 1980). The fact that learning patterns of the severely handicapped are often beset by slow rates of skill acquisition (Zeaman & House, 1979), deficits in long-term memory and recall ability (Ellis et al., 1982; Ellis & Woolridge, 1985; Glidden, 1979), and poor generalization skills (Stokes & Baer 1977) underscores the valuable role the laptop microcomputer can serve as an augmentative instructional tool.

The promise that these new-wave laptop computers hold for helping *all* handicapped learners acquire, maintain, and generalize functional skills has become a major topic of discussion. For example, in a major departure from trends of just a few years ago, many core textbooks in special education and communication disorders have begun to include sections on the use of microcomputer technology (e.g., Grossfeld & Grossfeld, 1986; Kirk & Gallagher, 1986; Meyen, 1982; Reynolds & Birch, 1982; Sanders, 1986; Schloss & Sedlak, 1986). At the same time, scores of new software titles are becoming available each year (LINC, 1986). In sum, the success of the microcomputer has been amply illustrated in many curriculum areas, including: (a) instruction and communication (Becker, 1986; Berninger, 1986; Cosden et al. 1987; Gardner & Brever, 1985; Green, in press; Parette, Strother & Hourcade, 1986; Rotheray, Sewell & Morton, 1986; Ruconich, Ashcroft & Young, 1986; Schwartz, 1984); (b) pure and applied research (Browning et al., 1986; Torgesen, 1986;

Woodward et al., 1986; Sheingold, Kane & Endreweit, 1983); (c) assessment instrumentation (Dattilo, 1986); Fitch et al., 1984); and (d) clinical treatment (Chandler, Burck, & Sampson, 1986; Katz, 1986; Lambert, Intrieri & Hollandsworth, 1986). Clearly, the arrival of the laptop has firmly established the computer as the fifth and perhaps most important feature of contemporary approaches to addressing and remediating a wide variety of learning handicaps.

Yet, concern abou the *quality* of *educational software* continues to pose a serious challenge to the technology (Hedbring, 1985; Mace, 1986). In fact, the flawed nature of educational softwrae has led to demands for tighter, more rigorous standards of "quality control" in the design of software targetted for all our schools (Bell, 1985; Caissy, 1984; Cohen, 1983; Duquette, 1985; Futrell & Geisert, 1985; Garner, 1982; Hedbring, 1985; Shanahan & Ryan, 1984; Test, 1985; Watt, 1984). Some writers have even gone so far as to label much of the current stock of educational software as a burgeoning "sea of junk" (e.g., Lewis, 1985). The question, of course, is just what sort of improvement is called for and how it can be properly accomplished.

A partial answer to improving the quality of educational software can be found at the grass roots level. Software purchasing personnel located in schools, clinics, and universities are in a powerful voting position to separate good empirical software from "vapidware" (Hedbring, 1985). All that is needed is a *practical*, brief, tightly-honed checklist. Software titles that failed to measure up would be summarily rejected; software earning high tallies on the checklist would be seriously considered for purchase. The result would not only be a more efficient software screening procedure but also a start message for software developers: publish high-grade software or perish at the purchasing counter.

One school-based computer center has instituted just such an effort. Called STEPPE-Lab, the center is a teaching-research laboratory for severely handicapped (communication-impaired, autistic, emotionally disturbed) learners located in New York City (Hedbring, 1985; Hedbring & Newsom, 1985). Faced with the task of screening many hundreds of software titles over the past three years, the Lab gradually developed a set of practical guidelines by which to justify eventual software purchasing decisions. The Software Evaluation Checklist presented below is the product of that effort. From an early list of more than a hundred criteria, the checklist was narrowed to twenty-five key items carefully ranked in terms of relative importance. The checklist is now used in a concerted and systematic attempt to avoid wasting precious dollars and shrinking software budgets on inferior, poorly designed, low-quality educational software.

The checklist is presented in four parts: First, the checklist items are ranked in terms of importance. Second, each item is accompanied by a comment section, including references to specific software products that exemplify a point discussed. Third, for convenience, the checklist is also presented in abbreviated from (Appendix A). Finally, Appendix B offers a list of several software programs that have scored high tallies on the STEPPE-Lab checklist. It should be noted that, all references to "the user" in the checklist refer to the generalized "educator" (academic teacher, language or communication therapist, speech therapist or pathologist, computer instructor, parent, et al.) rather than to the handicapped person or student.

SOFTWARE EVALUATION: A PRACTICAL CHECKLIST

1. *Compatibility:* Will the software run on your computer?

   The make and model of computer determines the compatibility of instructional software. Without special and expensive hardware, for example, Apple and IBM software are not interchangeable.

2. *Media:* Is the software available on 5.25" *and* 3.50" disks?

   Disks used to store instructional programs continue to shrink in size and price. Both IBM and Apple already have begun to move from the well-known 5.25 inch disk to the smaller, more reliable and greater capacity 3.50 inch format. As a result, software companies are beginning to offer their products in both media sizes, with the "AppleWorks" and "PFS" administrative packages serving as prime examples. In fact, as of late 1987, more than 250 popular software programs were available on the smaller disk format for Apple and IBM. Software that can be purchased on 3.50 inch micro-disks will save money when the computer lab is upgraded to accommodate these smaller and faster disk drives in the future.

3. *Backup disk:* Is an archival or backup disk of the software readily available?

   As compensation for copy protection schemes used to impede illegal disk-duplication, many companies offer at nominal cost a backup disk of their software products at the time of purchase. If an archival copy of the program is not available, the full cost of having to purchase a replacement disk in the future needs to be considered when budgeting for projected software acquisitions.

4. *Target population specified:* For which special education population is the program designed?

   Educators, clinicians, parents, and other potential software buyers are well aware of the functioning level and learning handicaps of persons expected to benefit from a software acquisition. Logically, then, these buyers need to know the reading level, mental age, or prerequisite skills the software program assumes or requires. Such information should be clearly stated in the manual; if not, beware! Try the software product before buying. Be certain the product will in fact be appropriate to the functioning level of the audience for whom it's intended.

5. *Returns policy:* Can unwanted software be returned?

   Can the software package be returned, if later it is found defective? Perhaps the program refuses to load, or maybe the manual omits crucial operating information. Is the warranty period 10, 30, 90 days? Are there circumstances under which the software can be returned after the warranty period has expired? In short, it is important to determine at the outset the extent to which the vendor is prepared to support the product.

6. *Cost:* What is the long-term cost of the software?

   There are a number of ancillary costs associated with buying educational software, including: (a) Backup cost: Is the archival or backup disk free? (b) Replacement cost: If the disk is damaged in use, is its replacement less expensive? (c) Returns cost: If a disk can be returned, is there a "restocking" fee? (d) Upgrades cost: What is the charge for obtaining an upgraded version of the program in the future? (e) Telephone support cost: Is a toll-free customer support telephone number available? (f) Documentation cost: If only one disk out of a multi-disk program set is purchased, is there an added cost for the accompanying instructions on how to use the program? In sum, some costs are obvious, others seem well-disguised.

7. *Empirical software:* Does the software accurately reflect the curriculum domain it purports to represent?

Teachers and clinicians who work with learning-handicapped or language-impaired persons are familiar with the many conditions of learning, principles of teaching, and rules of instruction that characterize specialized instructional environments. Unfortunately, many software developers do not possess this knowledge and their software thus suffers major flaws in design and pedagogy (Hedbring, 1985).

The extent to which the experimental literature is reflected in educational software can be referred to as software construct validity (after Cronbach, 1970, p. 125). If there is a high degree of construct validity, the software can be considered to be empirically-based and thus an appropriate instructional tool.

Two examples underscore the importance of identifying empirical software when making selection decisions. First, one purpose of a program entitled "Odd One Out" is to train squares and triangles. However, in order to be considered correct, a matching response must be made on the basis of colour, not shape! This requirement violates at least one well-known principle of instruction: "Train along the target dimension." That is, if shape is the stimulus dimension, train the shapeness of a stimulus, not a secondary dimension such as colour (Stevenson, 1972).

A second example involves a program called "First Letters & Words" that purports to teach shape and picture matching. Unfortunately, every one of the program's task frames consists of operating directions that require a higher level of cognitive development than the shape matching task itself! A second principle of effective curriculum design is thus violated: "Do not assume (as a prerequisite) the skill to be trained" (see Ball, 1971; Bruner, 1966; Dixon, 1981; Gibson, 1969; Handleman, Powers & Harris, 1984; Kephart, 1971; Montessori, 1964). Again, the program requires as prerequisites a set of skills more advanced developmentally than the task being taught! Clearly, software that is not empirical may violate basic learning principles.

8. *Load speed:* Does the software package load quickly?

Fast loading programs limit opportunities for off-task student behavior. For a teacher or clinician, there is little that is more disheartening than waiting even fifteen seconds for a blank screen to come to life with colour and sound. Programs slow to start invite disruptive pupil behavior. Hence, a practical rule of thumb is to avoid software that takes more than 30 seconds to load.

9. *Introductory credits:* Once loaded, does the software complete its menu of credits, logos, disclaimers, and other introductory screens quickly?

A major shortcoming of numerous software programs aimed at the special education market is the inordinate amount of time spent waiting for the software to run its course of introductory screens before the main menu appears. Thus, another useful rule is to look for software whose main menu appears within 60 seconds after the drive is turned on. Indeed, top-rated programs such as "Delta Drawing," "Printshop," "Magic Slate," and the "StickyBear" series all are ready for use in less than a minute.

10. *Attention screens:* Is the screen "busy" during memory swapping periods?

Computer screens tend to remain blank or "asleep" with many programs during periods a program segment is being moved in and out of memory. Inactive screens can lead to off-task and disruptive behaviour on the part of impatient learners. Thus, "distractor screens" help greatly to keep the user attentive during the loading process. Broderbund's "Printshop" is a fine example of a program that keeps the screen alive with visual activity while engaging in memory-related chores.

11. *Friendliness:* Is the software program easy to learn and easy to use?

Easy understanding leads to early and eager use. The more quickly a teacher or clinician is able to master a program, the more quickly the software will gain favor as an instructional or treatment tool. Before purchasing any program, then, try it out in the store or library and determine if its operation can be mastered within 30 minutes or so. Successful software, such as Delta Drawing, PrintShop, all DLM software, and most Random House offerings, can be mastered in a short period of time. On the other hand, Milliken and Hartley software, while of high quality, require relatively long learning curves.

12. *Failsafe operation:* Can the software be easily "crashed"?

Software that crashes, that "freezes", "hangs" or otherwise suddenly ceases to work is software that was not sufficiently tested during product development. There are few problems more frustrating during instruction than to be forced to suddenly curtail a lesson, load a different program, and resume instruction while at the same time trying to remember for future reference what caused the program to crash in the first place. Moreover, it's a time-tested truism that, when working with computers, any "bug" responsible for crashing a program will assuredly bite at the most inappropriate moment!

13. *Fault-tolerance:* Does the software check critical user input?

Software users often key-in software commands they immediately regret. For example, upon destroying a valuable student file with a misplaced keystroke, a teacher or clinician suddenly finds need for that very file after all. Too late! The software package accepted the input without question and summarily expunged the information! Simply stated, it's nice to have the luxury of changing a decision before a software command is entered in and executed.

Hence, look for strategically placed "Y/N" prompts or other error-checking cues such as "Are you sure?" "Really?" "Please enter the password", or "Please retype your answer". Such cues force a critical pause that can save time, effort, frustration, and valuable information before it's too late.

14. *On-line help:* When in trouble, is help but a key-stroke away?

Software users soon discover that practically every program has its own command structure that must be learned if the program is to be operated correctly. Over the course of the year, then, a teacher or clinician may be forced to learn a score of command structures in order to run newly acquired software. Having on-line operating instructions that can be invoked with a single keystroke reduces the need to commit to memory countless blocks of arcane commands. In short, software that offers on-line help is a major ally of the user!

15. *Flexible content:* Can the software content be modified?

Software that allows its content to be modified gives the teacher or clinician the freedom to match the program to the learning needs and functioning level of each handicapped person. "Canned" software that arrives preloaded with unalterable curriculum content at the very least removes from the user the right to determine what will be taught. Therefore, look for software that allows its content to be changed.

16. *One-pass introduction:* Does the software program bypass opening screens when it branches back to repeat a lesson?

Software should allow the user to repeat a lesson or module without the lengthy sequence of title, credit, and disclaimer screens with each program repetition. Hence, look for programs that link directly back to the lesson itself, thus avoiding the start-up or title screens.

17. *Program filters:* Can sections of the program be bypassed?

    Software should include "ESC" commands that allow the user to bypass program levels, software options, and operating directions. Accordingly, look for software that asks such questions as, "Do you want to see instructions?" "Do you want to keep score?" "Do you need help?" "Are you using a joystick?" "Do you want sound?"

18. *Task vs reward frames:* Are the task and reward frames of the program distinct and separate from one another?

    Numerous studies during the past fifteen years suggest that persons with learning handicaps often exhibit severe problems, delays, and deficits in attention, perception, and discrimination (e.g., Gibson, 1969; Hallahan & Reeve, 1980; Kierman, Jordan & Saunders, 1984; Kinsbourne & Caplan, 1979; Krupski, 1980; Stevenson, 1972; Trabasso, 1968; Zeaman & House, 1979). One lesson gleamed from these studies is the importance of designing instructional software that presents handicapped learners with crystal clear ("highly discriminable") correct versus incorrect response choices.

    A task screen should thus contain *only* the task, perhaps highlighted in white print and set against a dark background. The task screen should be uncluttered, totally free of colour, sound, animation, graphics, instructions and any other distractors that might interfere with the learner's need to attend to the task at hand. Combining both the task screen and the reward screen (as in the case of "First Letters and Words") floods the visual field with cues (colour, sound, animation, graphics) *not* relevant to solving the task itself. The result, as a wealth of studies have found, may be that the attention-deficient learner "locks onto" or gets "hooked by" the task-irrelevant information (see Bailey, 1981; Cook, Anderson & Rincover, 1982; Hedbring & Newsom, 1985; Schreibman, Koegel & Craig, 1977; Wilhelm & Lovaas, 1976). Consequently, the learner may never master the task, if the transition from the "prompt stimulus" (colour, graphics) to the "training stimulus" (the task to be learned) is not accomplished. It is this logic that presents a strong argument *against* "arcade-type" entertainment games. Such games, while trying to make drill and practice activities more imaginative and exciting, run the risk of inhibiting learning since task screens are typically flush with the fast-paced and animated graphics. Task and reward screens should be separate!

19. *Reward frames:* Are the reward frames varied, brief, multisensory?

    The literature on motivating handicapped learners stresses the importance of fast-paced, high-contract, sound-rich sensory rewards (e.g., Kierman, Jordan & Saunders, 1984; Lovaas, 1981; Murphy, 1982; Rincover et al., 1977). Therefore, look for simple, fast-paced, brightly-coloured, highly-animated reward frames that flash on the screen for five to ten seconds immediately subsequent to and contingent upon a correct response.

20. *Levels of difficulty:* Does the software program include multiple levels of task difficulty?

    Some programs offer many levels of difficulty (viz., "StickyBear Math"). The greater the number of levels of task difficulty that are included on a disk, the less disk-swapping will be required to accommodate the learner's progress (or retreat)

from one level of mastery to another. Moreover, if multiple copies of the program are purchased, the same program can be used with different students, with the difficulty level of each program copy set initially to match the functional status of each learner.

21. *Auto-branching:* Does the software program branch automatically among the multiple levels of task difficulty?

    Educational software that remains at the same level of the task difficulty until the teacher or clinician intervenes is a program that wastes time. Software not only offer several levels of difficulty, but also should automatically move between those levels as well. Otherwise, the teaching session is subjected to frequent and unnecessary disruption as the teacher or clinician manually changes the difficulty level of the task.

22. *On-line curriculum aids:* Are major components of traditional lesson plans included with the software?

    Good software designed for the special education market offers clearly stated instructional objectives that are prominently noted in the manual. Moreover, good software includes other instructional features, such as program prerequisites, behavioral objective, task analysis, suggested teaching procedures, and two or three relevant readings from professional journals (Hedbring & Holmes, 1977; Mager, 1975; Williams, Brown & Certo, 1975).

23. *Automatic record-keeping:* Does the program keep score?

    Software should offer the user an option to keep score. Not all instances of pupil performance necessarily call for data to be collected, of course. But the option should exist, along with additional options for charting, graphing, and printing out the results to several types of dot-matrix printers.

24. *Keyboard control:* Does the software program disable unused keys?

    There are numerous instances in which a majority of the keyboard keys are not used during an instructional task. There are many two-choice and match-to-sample exercises which require the student to press the key on the keyboard that is the same as the letter or number displayed in a box on the screen. With this task format, the number of student errors can be dramatically reduced if the software program disables all of the irrelevant keys ("stimulus cues").

    For example, if the letter in the box on the screen is an "A", an empirically based software program presents a brief reward screen flooded with colour, sound, graphics and motion. However, if *any* other key on the keyboard is pressed, nothing happens! And where nothing happens, extinction sets in! After a number of trials, the learner begins to scan the keys for the symbol that looks like the "thing" on the screen. Experience in STEPPE Lab is that appropriately locked out keyboard keys greatly facilitates two-choice discrimination and match-to-sample number and letter tasks.

25. *Alternative input options:* Can the software be used with a mouse? Joystick? Drawing tablet? Touch screen? Voice synthesizer?

    Increasingly, educational programs are becoming flexible in terms of input mode and communication device. Look for programs that can be used in conjunction with a mouse, joystick, drawing tablet, touchscreen, or voice synthesizer, as well as the more traditional keyboard. The more flexible the input options, the more appropriately the varying needs of learners with communication handicapped can be served.

SUMMARY

Jerrome Bruner (1960) offered a challenge years ago that merits renewed consideration amid today's advances. "Any subject", he argued, "can be taught in an intellectually honest form to any child at any stage of development" (p. 33). Given an instructional technology now bolstered by laptop microcomputers, only the lack of empirically pure and pedagogically correct software darkens the Brunerian dream. As concerned innovators, then, we will continue to apply the rigorous standards to software that we employ when considering all other factors dear to our children's education. The fact is that the present state of educational software leaves us no alternative.

ACKNOWLEDGEMENT

The author would like to thank the editor, Dr. Carl L. Parsons, for his valuable editorial comments during the preparation of this article.

Address all correspondence to Dr. Charles Hedbring, Program STEPPE, Suite 1712 Master Apartments, 310 Riverside Drive, New York, New York 10025, USA. (212) 866-6694.

REFERENCES

Adams, J.L. (1975). *An education curriculum for the moderately, severely, and profoundly mentally handicapped pupil.* Springfield, IL: Charles C. Thomas.

Anderson, D.R., Hodson, G.D. & Jones, W.G. (1975). *Instructional programming for the handicapped student.* Springfield, IL: Charles C. Thomas.

Bailey, S.L. (1981). Stimulus overselectivity in learning disabled children. *Journal of Applied Behavior Analysis, 14,* 239-248.

Ball, T.S. (1971). *Itard, Seguin, and Kephart: Sensory education — A learning interpretation.* Columbus, OH: Charles E. Merrill.

Becker, H.J. (1986). *Instructional uses of school computers.* Baltimore, MD: Johns Hopkins University.

Bell, M.E. (1985). The role of instructional theories in the evaluation of microcomputer courseware. *Educational Technology, 25(3),* 36-40.

Bender, M. & Valletutti, P.J. (1976). *Teaching the moderately and severely handicapped (Vols 1 & 2).* Baltimore, MD: University Park Press.

Berninger, V.W. (1986). Comparison of two microcomputer-assisted methods for teaching word decoding and encoding to non-vocal, non-writing and learning disabled students. *Programmed Learning and Educational Technology, 23,* 124-129.

Bijou, S.W. (1979). The technology of teaching young handicapped children. In S.W. Bijou and E. Ribes-Inesta (Eds.), *Behavior modification: Issues and extensions* (pp. 27-42). New York: Academic Press.

Bijou, S.W. (1979). Some clarifications on the meaning of behavior analysis of child development. *The Psychological Record, 29,* 3-13.

Bricker, W.A. & Bricker, D.D. (1974). An early language training strategy. In R.L. Schiefelbusch & J.L. Lloyd (Eds.), *Language perspectives: Acquisition, retardation, and intervention* (pp. 431-468). Baltimore: University Park Press.

Brown, L., Nietupski, J. & Hamre-Nietupski, S. (1976). Criterion of ultimate functioning. In M.A. Thomas (Ed.), *Hey, don't forget about me!* (pp. 2-15). Reston, VA: Council for Exceptional Children.

Browning, P., Nave, G., White, W. & Barkin, P. (1985). Interactive video as an instructional technology for handicapped learners: A development and research program. *Australia and New Zealand Journal of Developmental Disabilities, 11,* 123-128.

Bruner, J.S. (1960). *The process of education.* New York: Vintage Books.

Bruner, J.S. (1966). *Toward a theory of instruction.* New York: Norton.

Burns, E. (1985, July). The great arcade con. *Popular Computing,* p. 140.

Caissy, G.A. (1984). Evaluation educational software: A practitioner's guide. *Phi Delta Kappan, 66,* 249-250.

Chandler, G.M., Burck, H.D., & Sampson, J.P. (1986). A generic computer program for systematic desensitization: Description, construction, and case study. *Journal of Behavior Therapy and Experimental Psychiatry, 17,* 171-174.

Cook, A.R., Anderson, N. & Rincover, A. (1982). Stimulus overselectivity and stimulus control: Problems and strategies. In R.L. Koegel, A. Rincover & A.L. Egel (Eds.), *Educating and understanding autistic children* (pp. 90-105). San Diego: College-Hill Press.

Cosden, M A., Gerbder, M.M., Semmel, D.S., Goldman, S.R. & Semmel, M.I. (1987). Microcomputer use within micro-educational environments. *Exceptional Children, 53,* 399-409.

Cronbach, L.J. (1970). *Essentials of psychological testing* (3rd ed.). New York: Harper & Row.

Dattilo, J. (1986). Computerized assessment of preference for severely handicapped individuals. *Journal of Applied Behavior Analysis, 19,* 445-448.

Dixon, L.S. (1981). A functional analysis of photo-object matching skills of severely retarded adolescents. *Journal of Applied Behavior Analysis, 14,* 465-478.

Duquette, C. (1985). Formative evaluation of courseware: One instrument. *Educational Technology, 25(2),* 20-23.

Ellis, N.R. et al. (1982). Learning, memory, and transfer in profoundly, severely, and moderately mentally retarded persons. *American Journal of Mental Deficiency, 87,* 186-196.

Ellis, N.R., Deacon, J.R. & Wooldridge, P.W. (1985). Structural memory deficits of mentally retarded persons. *American Journal of Mental Deficiency, 89,* 393-402.

Engelmann, S. & Carnine, D. (1982). *Theory of instruction: Principles and applications.* New York: Irvington Publishers.

Fitch, K.L., Davis, L.A., Evans, W.B. & Sellers, D.E. (1984). Computer-managed screening for communication disorders. *Language, Speech, and Hearing Services in Schools, 15,* 66-69.

Fredericks, H.D. et al. (1976). *The teaching research curriculum for moderately and severely handicapped.* Springfield, IL: Charles C. Thomas.

Futrell, M. & Geisert, P. (1985). A call for action to improve the design of microcomputer instructional courseware. *Educational Technology, 25(5),* 13-15.

Gage, N.L. (1978). *The scientific basis of the art of teaching.* New York: Teachers College Press.

Gagne, R. N. (1977). *The conditions of learning* (3rd ed). New York: Holt, Rinehart, & Winston.

Garner, D. (1982). Educational microcomputer software: Nine questions to ask. *Science and Children, 19(6),* 24-25.

Garner, J.M. & Brever, A. (1985). Using microcomputers in the evaluation, training, and management of persons with developmental disabilities. *Australia and New Zealand Journal of Developmental Disabilities, 11,* 129-135.

Gibson, E.J. (1969). *Principles of perceptual learning and development.* New York: Appleton-Century-Crofts.

Glaser, R. (1976) Components of a psychology of instruction: Toward a science of design. *Review of Educational Research, 46*, 1–24.

Glidden, L.M. (1979). Training of learning and memory in retarded persons: Strategies, techniques, and teaching tools. In N.R. Ellis (Ed.), *Handbook of mental deficiency: Psychological theory and research (2nd ed)* (pp. 619–657). Hillsdale: NJ: Erlbaum Assoc.

Green, G. (1986). Micro-computer applications in speech pathology: Software availability. *Australian Journal of Human Communication Disorders, 14(2)*, 95–112.

Grossfeld, M.C. & Grossfeld, C.A. (1986). *Microcomputer applications in rehabilitation of communication disorders*. Rockville, Maryland: Aspen Publications.

Guess, D., Sailor, W. & Baer, D.M. (1976). *Functional speech and language training for the moderately and severely handicapped (Parts I & II)*. Lawrence, KS: H & H Enterprises.

Guess, D. & Siegel-Causey, E. (1985). Behavioral control and education of severely handicapped students: Who's doing what to whom? And Why? In D. Bricker & J. Filler (Eds.), *Severe mental retardation: From theory to practice* (pp. 230–244). Reston, VA: Council for Exceptional Children.

Hallahan, D.P. & Reeve, R.E. (1980). Selective attention and distractibility. In B. K. Keogh (Ed.), *Advances in special education* (pp. 141–181). Greenwich, CT: JAI Press.

Handleman, J.S., Powers, M.D. & Harris, S.L. (1984). Teaching of labels: An analysis of concrete and pictorial representations. *American Journal of Mental Deficiency, 88*, 625–629.

Haring, N. & Schiefelbusch, R.L. (1976). *Teaching special children*. New York: McGraw-Hill.

Harris, D. & Vanderheiden, G.C. (1980). Augmentative communication techniques. In R.L. Schiefelbusch (Ed.), *Nonspeech language and communication: Analysis and intervention* (pp. 259–301). Baltimore, MD: University Park Press.

Hedbring, C. (1985). Computers and autistic learners: An evolving technology. *Australian Journal of Human Communication Disorders, 13*, 169–194.

Hedbring, C. & Holmes, C. (1977). Getting it together with PL 94–142: The IEP in the classroom. *Education and Training of the Mentally Retarded, 12*, 212–224.

Hedbring, C. & Newsom, C. (1985). Visual overselectivity: A comparison of two instructional remediation procedures with autistic children. *Journal of Autism and Developmental Disorders, 15*, 9–22.

Katz, R.C. (1986). *Aphasia treatment and microcomputers*. London: Taylor & Francis.

Kazdin, A.E. (1978). *History of behavior modification*. Baltimore, MD: University Park Press.

Kent, L.R. (1974). *Language acquisition program for the retarded or multiply impaired*. Champaign, IL: Research Press.

Kephart, N.C. (1971). *The slow learner in the classroom* (2nd ed.). Columbus, OH: Charles E. Merrill.

Kierman, C., Jordan, R. & Saunders, C. (1984). *Stimulating the exceptional child*. Englewood Cliffs, NJ: Prentice-Hall.

Kinsbourne, M. & Caplan, P.J. (1979). *Children's learning and attention problems*. Boston: Little, Brown, & Co.

Kirk, S.A. & Gallagher, J.J. (1986). *Educating exceptional children* (5th ed.). Boston: Houghton-Mifflin.

Kissinger, E.M. (1981). A sequential curriculum for the severely and profoundly mentally retarded/multi-handicapped. Springfield, IL: Charles C. Thomas.

Krupski, A. (1980). Attention processes: Research, theory, and implications for special education. In B.K. Keogh (Ed.), *Advances in special education* (Vol. 1) (pp. 101–140). Greenwich, CT: JAI Press.

Lambert, M.E., Intrieri, R.C. & Hollandsworth, (1986). Development of a computerized reference retrieval system: A behavior therapy training tool. *Journal of Behavior Therapy and Experimental Psychiatry, 17*, 167–169.

Lewis, P.H. (1985, September 24). Fishing for gems of educational software in a sea of junk. *The New York Times*, p. C7.

Linc, (1986). *The specialware directory*. Columbus, OH: Author.

Lovaas, O.I. (1981). *Teaching developmentally disabled children*. Baltimore, MD: University Park Press.

Maze, S. (1986, April 14). Educators urge better software instruction. *InfoWorld*, p. 5

Mager, R. R. (1975). *Preparing instructional objectives* (2nd ed.). Belmont, CA: Fearon, Inc.

Meyen, E.L. (1982). *Exceptional children in today's schools: An alternative resource book*. Denver: Love Publishing.

Montessori, M. (1964). *The Montessori method*. New York: Schocken Books.

Murphy, G. (1982). Sensory reinforcement in the mentally handicapped and autistic child: A review. *Journal of Autism and Developmental Disorders, 12*, 265–278.

Parette, H.P., Strother, P.O. & Hourcade, J.J. (1986). Microswitches and adaptive equipment for severely impaired students. *Teaching Exceptional Chhnildren, 19(1)*, 15–18.

Popovich, D. (1977). *A prescriptive behavioral checklist for the severely and profoundly retarded*. Baltimore: University Park Press.

Popovich, D. (1981). *Effective educational and behavioral programming for severely and profoundly handicapped students*. Baltimore: Paul Brookes.

Popovich, D. & Laham, S.L. (1981). *The adaptive behavior curriculum*. Baltimore: Paul Brookes.

Reynolds, M.C. & Birch, J.W. (1982). *Teaching exceptional children in all America's schools* (rev. ed.). Reston, VA: Council for Exceptional Children.

Rincover, A., Newsom, C.D., Lovaas, O.I. & Koegel, R.L. (1977). Some motivational properties of sensory reinforcement in psychotic children. *Journal of Experimental Child Psychology, 24*, 312–323.

Rotheray, D.R., Sewell, D.F. & Morton, J.R. (1986). The design of educational software for children with severe learning difficulties. *Programmed Learning and Educational Technology, 23*, 119–123.

Ruconich, S.K., Ashcroft, S.C. & Young, M.F. (1986). Making microcomputers accessible to blind persons. *Journal of Special Education Technology, 8*, 27–46.

Sailor, W., Guess, D., Goetz, L., Schuler, A., Utley, B. & Baldwin, M. (1980). Language and severely handicapped persons: Deciding what to teach to whom. In W. Sailor, B. Wilcox & L. Brown (Eds.), *Methods of instruction for severely handicapped students* (pp. 71–105). Baltimore: Paul Brookes.

Sailor, W., Wilcox, B. & Brown, L. (Eds.). (1980). *Methods of instruction for severely handicapped students* (pp. 71–105). Baltimore: Paul Brookes.

Sanders, J. (1985). *Microcomputer applications for speech-language services in the schools*. London: Taylor & Francis.

Schloss, P.J. & Sedlak, R.A. (1986). *Instructional methods for students with learning and behavior problems*. Boston: Allyn & Bacon.

Schreibman, L., Koegel, R.L. & Craig, M.S. (1977). Reducing stimulus overselectivity in autistic children. *Journal of Abnormal Child Psychology, 5*, 425–436.

Schwartz, A.H. (Ed.). (1984). *Handbook of microcomputer applications in communication disorders*. San Diego: College Hill Press.

Shanahan, D. & Ryan, A.W. (1984). A tool for evaluating educational software. *Teaching Exceptional Children, 16(4)*, 242–247.

Sheingold, K., Kane, J.H. & Endreweit, M.E. (1983). Microcomputer use in schools: Developing a research agenda. *Harvard Educational Review, 53,* 412–432.

Snell, M.E. (Ed.). (1983). *Systematic instruction of the moderately and severely handicapped* (2nd ed). Columbus, OH: Charles E. Merrill.

Sontag, E. (Ed.). (1977). *Educational programming for the severely and profoundly retarded.* Reston, VA: Council for Exceptional Children.

Stevenson, H.W. (1972). *Children's learning.* Englewood Cliffs, NJ: Prentice-Hall.

Stokes, T.R. & Baer, D.M. (1977). An implicit technology of generalization. *Journal of Applied Behaviour Analysis, 10,* 349–367.

Taba, H. (1962). *Curriculum development: Theory and practice.* New York: Harcourt & Brace.

Tyler, R.W. (1949). *Basic principles of curriculum and instruction.* Chicago: University of Chicago Press.

Tawney, J.W. (1979). *The programmed environments curriculum.* Columbus, OH: Charles E. Merrill.

Test, D.W. (1985). Evaluating education software for the microcomputer. *Journal of Special Education Technology, 7(1),* 37–46.

Torgensen, J.K. (1986). Computers and cognition in reading: A focus on decoding fluency. *Exceptional Children, 53,* 157–162.

Trabasso, T. (1968, October). Pay attention. *Psychology Today,* pp. 30–36.

VORT Corporation. (1980). *Behavior Chart Progression (BCP) Methods Books.* Palo Alto, CA: Author.

Watt, D. (1984, July). Computer evaluation cometh. *Popular Computing,* 91–94.

White, O.R. (1980). Adaptive performance objectives. In W. Sailor, B. Wilcox and Brown, L. (Eds.), *Methods of instruction for severely handicapped students* (pp. 47–69). Baltimore, MD: Paul Brookes.

Wilhelm, H. & Lovaas, O.I. (1976). Stimulus overselectivity: A common feature in autism and mental retardation. *American Journal of Mental Deficiency, 81,* 26–31.

Williams, W., Brown, L. & Certo, N. (1975). Basic commponents of instructional programs. *Theory Into Practice, 14,* 123–136.

Woodward, J., Carnine, D., Gersten, R., Gleason, M., Johnson, G. & Collins, M. (1986). Applying instructional design principles to CAI for mildly handicapped students: Four recently conducted studies. *Journal of Special Education Technology, 8,* 13–26.

Zeaman, D. & House, B.J. (1979). The role of attention in retardate discrimination learning. In N.R. Ellis (Ed.), *Handbook of mental deficiency* (2nd ed., pp. 159–223). New York: McGraw-Hill.

APPENDIX A

PRACTICAL SOFTWARE EVALUATION CHECKLIST
Program STEPPE
Suite 1712 Master Apts.
310 Riverside Drive
New York, NY 10025

1. *Compatibility:* Will the software run on your computer?
2. *Media:* Is the software available on 5.25" *and* 3.50" disks?.
3. *Backup disk:* Is an archival or backup disk of the software readily available?
4. *Target population specified:* For which special education population is the program designed?
5. *Returns policy:* Can unwanted software be returned?
6. *Cost:* What is the long-term cost of the software?
7. *Empirical software:* Does the software accurately reflect the curriculum domain it purports to represent?
8. *Load speed:* Does the software package load quickly.
9. *Introductory credits:* Once loaded, does the software complete its menu of credits, logos, disclaimers, and other introductory screens quickly?
10. *Attention screens:* Is the screen "busy" during memory swapping periods?
11. *Friendliness:* Is the software program easy to learn and easy to use?
12. *Failsafe operation:* Can the software be easily "crashed"?
13. *Fault-tolerance:* Does the software check critical user input?
14. *On-Line help:* When in trouble, is help but a key-stroke away?
15. *Flexible content:* Can the software content be modified?
16. *One-pass introduction:* Does the software program bypass opening screens when it branches back to repeat a lesson?
17. *Program filters:* Can sections of the program be bypassed?
18. *Task vs. Reward Frames:* Are the task and reward frames of the program distinct and separate from one another?
19. *Reward frames:* Are the reward frames varied, brief, multisensory?
20. *Levels of difficulty:* Does the software program include multiple levels of task difficulty?
21. *Auto-branching:* Does the software program branch automatically among the multiple levels of task difficulty?
22. *On-line curriculum aids:* Are major components of traditional lesson plans included with the software?
23. *Automatic record-keeping:* Does the program keep score?
24. *Keyboard control:* Does the software program disable unnecessary keys?
25. *Alternative input options:* Can the software be used with a joystick? Mouse? Drawing tablet? Touch screen? Voice synthesizer?

APPENDIX B

RECOMMENDED SOFTWARE: APPLE

Baudville
- Blazing Paddles

Broderbund Software
- Dazzle Draw
- Print Shop series
- Where in the World is Carmen San Diego

DLM, Inc.
- Alphabet Circus
- Number Farm
- Shape & Color Rodeo
- Word Radar

Davidson & Associates, Inc.
- Word Attack!

Hartley Courseware, Inc.
- Clock
- Letter Recognition
- Memory Match
- Who-What-Where-When-Why
- Grammar Series

Iostens Learning Corp.
(Ufonic Voice Synthesizer)
- A Treasure Hunt of Facts
- Alphabet Academy
- Speech Composer
- Spelling Attack

Laureate Learning Systems
- Micro-LADS

Mindscape
- Crossword Magic
- SongWriter

Program STEPPE
- FACTS +

Random House
- Charlie Brown's ABC's
- Charlie Brown's 1,2,3's
- Peanuts Picture Puzzlers
- Snoopy Writer
- Snoopy Word Scrambler
- Garfield Double Dares

Spinnaker, Inc.
- Certificate Maker
- Delta Drawing
- Facemaker

Sunburst Communications, Inc.
- Factory
- Gnee series
- Magic Slate
- Teddy's Playground
- TouchWindow hardware +
- Touch 'N Write

Zerox/Weekly Reader Software
- StickyBear ABC
- StickyBear Math
- StickyBear Numbers
- StickyBear Shapes

RECOMMENDED SOFTWARE: IBM

Alphabet Circus
Blazing Paddles
Certificate Maker
Factory
Newsroom
Gnee or Not Gnee
Iggy's Gnees
Movie Maker
PFS: File
Printshop series
Program STEPPE Facts +
Rocky's Boots
StickyBear Math, Numbers
Where in the World is Carmen San Diego
DLM: Alphabet Circus, Number Farm, Shape & Color Rodeo, Word Radar.